Chapter 14
Erasures and Error-Correcting Codes

14.1 Introduction

It is well known that an (n, k, d_{min}) error-correcting code $\mathscr{C}$, where n and k denote the code length and information length, can correct $d_{min} - 1$ erasures [15, 16] where d_{min} is the minimum Hamming distance of the code. However, it is not so well known that the average number of erasures correctable by most codes is significantly higher than this and almost equal to $n - k$. In this chapter, an expression is obtained for the probability density function (PDF) of the number of correctable erasures as a function of the weight enumerator function of the linear code. Analysis results are given of several common codes in comparison to maximum likelihood decoding performance for the binary erasure channel. Many codes including BCH codes, Goppa codes, double-circulant and self-dual codes have weight distributions that closely match the binomial distribution [13–15, 19]. It is shown for these codes that a lower bound of the number of correctable erasures is $n-k-2$. The decoder error rate performance for these codes is also analysed. Results are given for rate 0.9 codes and it is shown for code lengths 5000 bits or longer that there is insignificant difference in performance between these codes and the theoretical optimum maximum distance separable (MDS) codes. The results for specific codes are given including BCH codes, extended quadratic residue codes, LDPC codes designed using the progressive edge growth (PEG) technique [12] and turbo codes [1].

The erasure correcting performance of codes and associated decoders has received renewed interest in the study of network coding as a means of providing efficient computer communication protocols [18]. Furthermore, the erasure performance of LDPC codes, in particular, has been used as a measure of predicting the code performance for the additive white Gaussian noise (AWGN) channel [6, 17]. One of the first analyses of the erasure correction performance of particular linear block codes is provided in a key-note paper by Dumer and Farrell [7] who derive the erasure correcting performance of long binary BCH codes and their dual codes. Dumer and Farrell show that these codes achieve capacity for the erasure channel.

M. Tomlinson et al., *Error-Correction Coding and Decoding*,
Signals and Communication Technology,
DOI 10.1007/978-3-319-51103-0_14

14.2 Derivation of the PDF of Correctable Erasures

14.2.1 Background and Definitions

A set of s erasures is a list of erased bit positions defined as f_i where

$$0 < i < s \quad f_i \in 0 \ldots n-1$$

A codeword $\mathbf{x} = x_0, x_1 \ldots x_{n-1}$ satisfies the parity-check equations of the parity-check matrix $\mathbf{H}$

$$\mathbf{H}\,\mathbf{x}^{\mathbf{T}} = \mathbf{0}$$

A codeword with s erasures is defined as

$$\mathbf{x} = (x_{u_0}, x_{u_1} \ldots x_{u_{n-1-s}} | x_{f_0}, x_{f_1} \ldots x_{f_{s-1}})$$

where x_{u_j} are the unerased coordinates of the codeword, and the set of s erased coordinates is defined as $\mathbf{f_s}$. There are a total of $n-k$ parity check equations and provided the erased bit positions correspond to independent columns of the $\mathbf{H}$ matrix, each of the erased bits may be solved using a parity-check equation derived by the classic technique of Gaussian reduction [15–17]. For maximum distance separable (MDS) codes, [15], any set of s erasures are correctable by the code provided that

$$s \leq n-k \tag{14.1}$$

Unfortunately, the only binary MDS codes are trivial codes [15].

14.2.2 The Correspondence Between Uncorrectable Erasure Patterns and Low-Weight Codewords

Provided the code is capable of correcting the set of s erasures, then a parity-check equation may be used to solve each erasure, viz:

$$\begin{array}{lll}
x_{f_0} = h_{0,0}x_{u_0} & +h_{0,1}x_{u_1} + h_{0,2}x_{u_2} & + \ldots h_{0,n-s-1}x_{u_{n-s-1}} \\
x_{f_1} = h_{1,0}x_{u_0} & +h_{1,1}x_{u_1} + h_{1,2}x_{u_2} & + \ldots h_{1,n-s-1}x_{u_{n-s-1}} \\
x_{f_2} = h_{2,0}x_{u_0} & +h_{2,1}x_{u_1} + h_{2,2}x_{u_2} & + \ldots h_{2,n-s-1}x_{u_{n-s-1}} \\
\ldots & \ldots\ldots & \ldots\ldots \\
x_{f_{s-1}} = h_{s-1,0}x_{u_0} & +h_{s-1,1}x_{u_1} + h_{s-1,2}x_{u_2} & + \ldots h_{s-1,n-s-1}x_{u_{n-s-1}}
\end{array}$$

where $h_{i,j}$ is the coefficient of row i and column j of $\mathbf{H}$.

As the parity-check equations are Gaussian reduced, no erased bit is a function of any other erased bits. There will also be $n-k-s$ remaining parity-check equations, which do not contain any of the erased bits' coordinates x_{f_j}:

$$h_{s,0}x_{u_0} + h_{s,1}x_{u_1} + h_{s,2}x_{u_2} + \cdots + h_{s,n-s-1}x_{u_{n-s-1}} = 0$$

$$h_{s+1,0}x_{u_0} + h_{s+1,1}x_{u_1} + h_{s+1,2}x_{u_2} + \cdots + h_{s+1,n-s-1}x_{u_{n-s-1}} = 0$$

$$h_{s+2,0}x_{u_0} + h_{s+2,1}x_{u_1} + h_{s+2,2}x_{u_2} + \cdots + h_{s+2,n-s-1}x_{u_{n-s-1}} = 0$$

$$\cdots$$

$$\cdots$$

$$h_{n-k-1,0}x_{u_0} + h_{n-k-1,1}x_{u_1} + h_{n-k-1,2}x_{u_2} + \cdots + h_{n-k-1,n-s-1}x_{u_{n-s-1}} = 0$$

Further to this, the hypothetical case is considered where there is an additional erased bit x_{f_s}. This bit coordinate is clearly one of the previously unerased bit coordinates, denoted as x_{u_p}.

$$x_{f_s} = x_{u_p}$$

Also, in this case it is considered that these $s+1$ erased coordinates do not correspond to $s+1$ independent columns of the $\mathbf{H}$ matrix, but only to $s+1$ dependent columns. This means that x_{u_p} is not contained in any of the $n-k-s$ remaining parity-check equations, and cannot be solved as the additional erased bit.

For the first s erased bits whose coordinates do correspond to s independent columns of the $\mathbf{H}$ matrix, the set of codewords is considered in which all of the unerased coordinates are equal to zero except for x_{u_p}. In this case the parity-check equations above are simplified to become:

$$\begin{aligned} x_{f_0} &= h_{0,p}x_{u_p} \\ x_{f_1} &= h_{1,p}x_{u_p} \\ x_{f_2} &= h_{2,p}x_{u_p} \\ \ldots &= \ldots \\ \ldots &= \ldots \\ x_{f_{s-1}} &= h_{s-1,p}x_{u_p} \end{aligned}$$

As there are, by definition, at least $n-s-1$ zero coordinates contained in each codeword, the maximum weight of any of the codewords above is $s+1$. Furthermore, any erased coordinate that is zero may be considered as an unsolved coordinate, since no non-zero coordinate is a function of this coordinate. This leads to the following theorem.

Theorem 1 *The non-zero coordinates of a codeword of weight w that is not the juxtaposition of two or more lower weight codewords, provide the coordinate positions of* $w-1$ *erasures that can be solved and provide the coordinate positions of w erasures that cannot be solved.*

Proof The coordinates of a codeword of weight w must satisfy the equations of the parity-check matrix. With the condition that the codeword is not constructed from the juxtaposition of two or more lower weight codewords, the codeword must have $w-1$ coordinates that correspond to linearly independent columns of the **H** matrix and w coordinates that correspond to linearly dependent columns of the **H** matrix.

Corollary 1 *Given s coordinates corresponding to an erasure pattern containing s erasures,* $s \leq (n-k)$*, of which w coordinates are equal to the non-zero coordinates of a single codeword of weight w, the maximum number of erasures that can be corrected is* $s-1$ *and the minimum number that can be corrected is* $w-1$.

Corollary 2 *Given* $w-1$ *coordinates that correspond to linearly independent columns of the* **H** *matrix and w coordinates that correspond to linearly dependent columns of the* **H** *matrix, a codeword can be derived that has a weight less than or equal to w.*

The weight enumeration function of a code [15] is usually described as a homogeneous polynomial of degree n in x and y.

$$W_{\mathscr{C}}(x,y) = \sum_{i=0}^{n-1} A_i x^{n-i} y^i$$

The support of a codeword is defined [15] as the coordinates of the codeword that are non-zero. The probability of the successful erasure correction of s or more erasures is equal to the probability that no subset of the s erasure coordinates corresponds to the support of any codeword.

The number of possible erasure patterns of s erasures of a code of length n is $\binom{n}{s}$. For a *single* codeword of weight w, the number of erasure patterns with s coordinates that include the support of this codeword is $\binom{n-w}{s-w}$. Thus, the probability of a subset of the s coordinates coinciding with the support of a single codeword of weight w, $prob(\mathbf{x_w} \in \mathbf{f_s})$ is given by:

$$prob(\mathbf{x_w} \in \mathbf{f_s}) = \frac{\binom{n-w}{s-w}}{\binom{n}{s}}$$

and

$$prob(\mathbf{x_w} \in \mathbf{f_s}) = \frac{(n-w)!\, s!\, (n-s)!}{n!\, (s-w)!\, (n-s)!}$$

simplifying

$$prob(\mathbf{x_w} \in \mathbf{f_s}) = \frac{(n-w)!\, s!}{n!\, (s-w)!}$$

In such an event the s erasures are uncorrectable because, for these erasures, there are not s independent parity-check equations [15, 16]. However, $s-1$ erasures are correctable provided the $s-1$ erasures do not contain the support of a lower weight codeword.

The probability that s erasures will contain the support of at least one codeword of any weight, is upper and lower bounded by

$$1-\prod_{j=d_{min}}^{s}1-A_j\frac{(n-j)!s!}{n!(s-j)!}<P_s\leq\sum_{j=d_{min}}^{s}A_j\frac{(n-j)!s!}{n!(s-j)!} \tag{14.2}$$

And given $s+1$ erasures, the probability that exactly s erasures are correctable, $Pr(s)$ is given by

$$Pr(s)=P_{s+1}-P_s \tag{14.3}$$

Given up to $n-k$ erasures the average number of erasures correctable by the code is

$$\overline{N_e}=\sum_{s=d_{min}}^{n-k}sPr(s)=\sum_{s=d_{min}}^{n-k}s\left(P_{s+1}-P_s\right). \tag{14.4}$$

Carrying out the sum in reverse order and noting that $P_{n-k+1}=1$, the equation simplifies to become

$$\overline{N_e}=(n-k)-\sum_{s=d_{min}}^{n-k}P_s \tag{14.5}$$

An MDS code can correct $n-k$ erasures and is clearly the maximum number of correctable erasures as there are only $n-k$ independent parity-check equations. It is useful to denote an MDS shortfall

$$\text{MDS}_{\text{shortfall}}=\sum_{s=d_{min}}^{n-k}P_s \tag{14.6}$$

and

$$\overline{N_e}=(n-k)-\text{MDS}_{\text{shortfall}} \tag{14.7}$$

with

$$\sum_{s=d_{min}}^{n-k}1-\prod_{j=d_{min}}^{s}1-A_j\frac{(n-j)!s!}{n!(s-j)!}<\text{MDS}_{\text{shortfall}} \tag{14.8}$$

and

$$\text{MDS}_{\text{shortfall}} < \sum_{s=d_{min}}^{n-k} \sum_{j=d_{min}}^{s} A_j \frac{(n-j)!s!}{n!(s-j)!} \tag{14.9}$$

The contribution made by the high multiplicity of low-weight codewords to the shortfall in MDS performance is indicated by the probability $\hat{P}_j$ that the support of at least one codeword of weight j is contained in s erasures averaged over the number of uncorrectable erasures s, from $s = d_{min}$ to $n-k$, and is given by

$$\hat{P}_j = \sum_{s=d_{min}}^{n-k} Pr(s-1) A_j \frac{(n-j)!s!}{n!(s-j)!} \tag{14.10}$$

14.3 Probability of Decoder Error

For the erasure channel with erasure probability p, the probability of codeword decoder error, $P_d(p)$ for the code may be derived in terms of the weight spectrum of the code assuming ML decoding. It is assumed that a decoder error is declared if more than $n-k$ erasures occur and that the decoder does not resort to guessing erasures. The probability of codeword decoder error is given by the familiar function of p.

$$P_d(p) = \sum_{s=1}^{n} P_s p^s (1-p)^{(n-s)} \tag{14.11}$$

Splitting the sum into two parts

$$P_d(p) = \sum_{s=1}^{n-k} P_s p^s (1-p)^{(n-s)} + \sum_{s=n-k+1}^{n} P_s p^s (1-p)^{(n-s)} \tag{14.12}$$

The second term gives the decoder error rate performance for a hypothetical MDS code and the first term represents the degradation of the code compared to an MDS code. Using the upper bound of Eq. (14.2),

$$\begin{aligned} P_d(p) \leq & \sum_{s=1}^{n-k} \sum_{j=1}^{s} A_j \frac{(n-j)!\ s!}{n!\ (s-j)!} \frac{n!}{(n-s)!\ s!} p^s (1-p)^{(n-s)} \\ & + \sum_{s=n-k+1}^{n} \frac{n!}{(n-s)!\ s!} p^s (1-p)^{(n-s)} \end{aligned} \tag{14.13}$$

As well as determining the performance shortfall, compared to MDS codes, in terms of the number of correctable erasures it is also possible to determine the loss from capacity for the erasure channel. The capacity of the erasure channel with erasure probability p was originally determined by Elias [9] to be $1 - p$. Capacity may be approached with zero codeword error for very long codes, even using non-MDS codes such as BCH codes [7]. However, short codes and even MDS codes, will produce a non-zero frame error rate (FER). For $(n, k, n - k + 1)$ MDS codes, a codeword decoder error is deemed to occur whenever there are more than $n - k$ erasures. (It is assumed here that the decoder does not resort to guessing erasures that cannot be solved). This probability, $P_{MDS}(p)$, is given by

$$P_{MDS}(p) = 1 - \sum_{s=0}^{n-k} \frac{n!}{(n-s)!\, s!} p^s (1-p)^{(n-s)} \tag{14.14}$$

The probability of codeword decoder error for the code may be derived from the weight enumerator of the code using Eq. (14.13).

$$P_{code}(p) = \sum_{s=d_{min}}^{n-k} \sum_{j=d_{min}}^{s} \left(A_j \frac{(n-j)!\, s!}{n!\, (s-j)!} \frac{n!}{(n-s)!\, s!} p^s (1-p)^{(n-s)} + \sum_{s=n-k+1}^{n} \frac{n!}{(n-s)!\, s!} p^s (1-p)^{(n-s)} \right) \tag{14.15}$$

This simplifies to become

$$P_{code}(p) = \sum_{s=d_{min}}^{n-k} \sum_{j=d_{min}}^{s} A_j \frac{(n-j)!\, (n-s)!}{(s-j)!} p^s (1-p)^{(n-s)} + P_{MDS}(p) \tag{14.16}$$

The first term in the above equation represents the loss from MDS code performance.

14.4 Codes Whose Weight Enumerator Coefficients Are Approximately Binomial

It is well known that the distance distribution for many linear, binary codes including BCH codes, Goppa codes, self-dual codes [13–15, 19] approximates to a binomial distribution. Accordingly,

$$A_j \approx \frac{n!}{(n-j)!\, j!\, 2^{n-k}} \tag{14.17}$$

For these codes, for which the approximation is true, the shortfall in performance

compared to an MDS code, $MDS_{shortfall}$ is obtained by substitution into Eq. (14.9)

$$MDS_{shortfall} = \sum_{s=1}^{n-k} \sum_{j=1}^{s} \frac{n!}{(n-j)!\, j!\, 2^{n-k}} \frac{(n-j)!\, s!}{n!\, (s-j)!} \tag{14.18}$$

which simplifies to

$$MDS_{shortfall} = \sum_{s=1}^{n-k} \frac{2^s - 1}{2^{n-k}} \tag{14.19}$$

which leads to the simple result

$$MDS_{shortfall} = 2 - \frac{n-k-2}{2^{n-k}} \approx 2 \tag{14.20}$$

It is apparent that for these codes the MDS shortfall is just 2 bits from correcting all $n - k$ erasures. It is shown later using the actual weight enumerator functions for codes, where these are known, that this result is slightly pessimistic since in the above analysis there is a non-zero number of codewords with distance less than d_{min}. However, the error attributable to this is quite small. Simulation results for these codes show that the actual MDS shortfall is closer to 1.6 bits due to the assumption that there is never an erasure pattern which has the support of more than one codeword.

For these codes whose weight enumerator coefficients are approximately binomial, the probability of the code being able to correct exactly s erasures, but no more, may also be simplified from (14.2) and (14.3).

$$\begin{aligned} Pr(s) = &\sum_{j=1}^{s+1} \frac{n!}{(n-j)!\, j!\, 2^{n-k}} \frac{(n-j)!\, (s+1)!}{n!\, (s+1-j)!} \\ &- \sum_{j=1}^{s} \frac{n!}{(n-j)!\, j!\, 2^{n-k}} \frac{(n-j)!\, s!}{n!\, (s-j)!} \end{aligned} \tag{14.21}$$

which simplifies to become

$$Pr(s) = \frac{2^s - 1}{2^{n-k}} \tag{14.22}$$

for $s < n - k$ and for $s = n - k$

$$Pr(n-k) = 1 - \sum_{j=1}^{n-k} \frac{n!}{(n-j)!\, j!\, 2^{n-k}} \frac{(n-j)!\, (n-k)!}{n!\, (n-k-j)!} \tag{14.23}$$

and

Table 14.1 PDF of number of correctable erasures for codes whose weight enumerator coefficients are binomial

Correctable erasures	Probability
$n-k$	$\frac{1}{2^{n-k}}$
$n-k-1$	$0.5-\frac{1}{2^{n-k}}$
$n-k-2$	$0.25-\frac{1}{2^{n-k}}$
$n-k-3$	$0.125-\frac{1}{2^{n-k}}$
$n-k-4$	$0.0625-\frac{1}{2^{n-k}}$
$n-k-5$	$0.03125-\frac{1}{2^{n-k}}$
$n-k-6$	$0.0150625-\frac{1}{2^{n-k}}$
$n-k-7$	$0.007503125-\frac{1}{2^{n-k}}$
$\vdots$	$\vdots$
$n-k-s$	$\frac{1}{2^s}-\frac{1}{2^{n-k}}$

$$Pr(n-k)=\frac{1}{2^{n-k}} \tag{14.24}$$

For codes whose weight enumerator coefficients are approximately binomial, the pdf of correctable erasures is given in Table 14.1.

The probability of codeword decoder error for these codes is given by substitution into (14.15),

$$P_{code}(p)=\sum_{s=0}^{n-k}\left(\frac{2^s-1}{2^{n-k}}\right)\frac{n!}{(n-s)!\,s!}p^s(1-p)^{(n-s)}+P_{MDS}(p) \tag{14.25}$$

As first shown by Dumer and Farrell [7] as n is taken to ∞, these codes achieve the erasure channel capacity. As examples, the probability of codeword decoder error for hypothetical rate 0.9 codes, having binomial weight distributions, and lengths 100 to 10,000 bits are shown plotted in Fig. 14.1 as a function of the channel erasure probability expressed in terms of relative erasure channel capacity $\frac{0.9}{1-p}$. It can be seen that at a decoder error rate of 10^{-8} the (1000, 900) code is operating at 95% of channel capacity, and the (10,000, 9,000) code is operating at 98% of channel capacity. A comparison with MDS codes is shown in Fig. 14.2. For codelengths from 500 to 50,000 bits, it can be seen that for codelengths of 5,000 bits and above, these rate 0.9 codes are optimum since their performance is indistinguishable from the performance of MDS codes with the same length and rate.

A comparison of MDS codes to codes with binomial weight enumerator coefficients is shown in Fig. 14.3 for $\frac{1}{2}$ rate codes with code lengths from 128 to 1024.

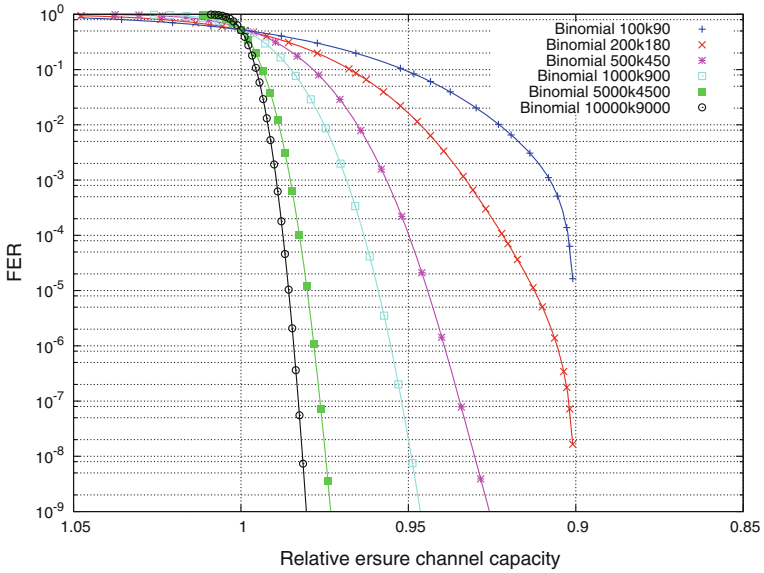

Fig. 14.1 FER performance of codes with binomial weight enumerator coefficients

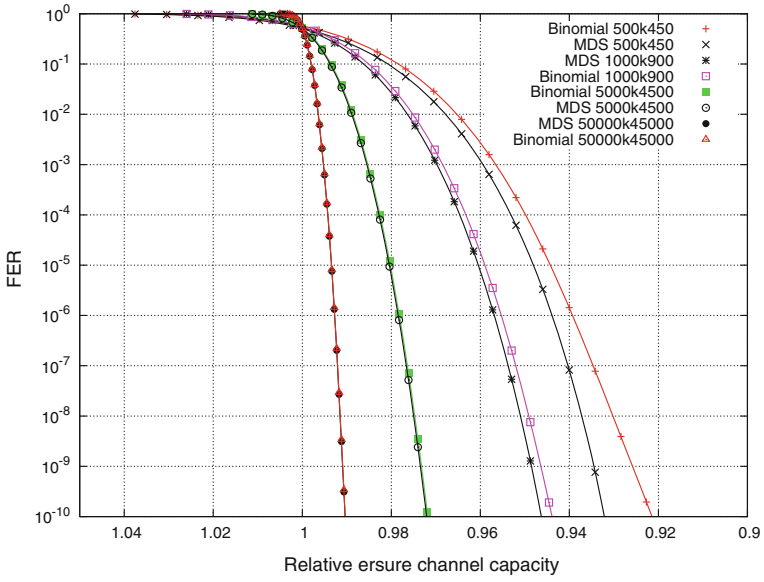

Fig. 14.2 Comparison of codes with binomial weight enumerator coefficients to MDS codes

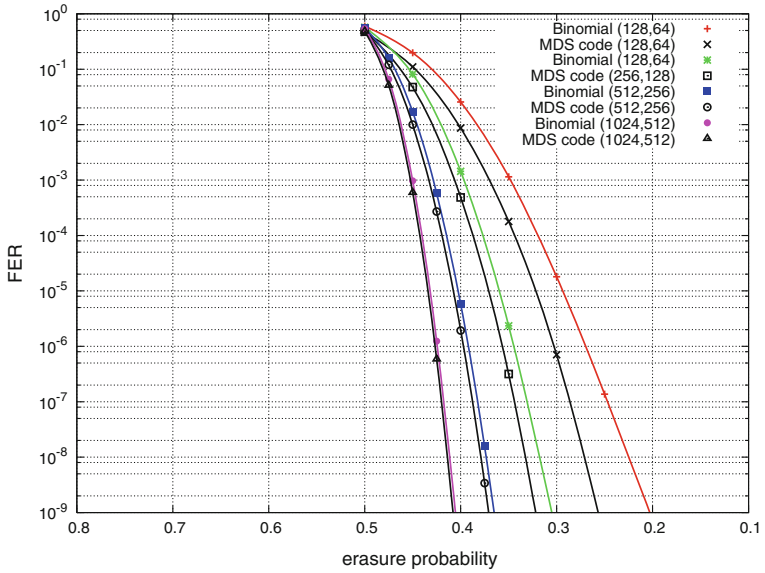

Fig. 14.3 Comparison of half rate codes having binomial weight enumerator coefficients with MDS codes as a function of erasure probability

14.5 MDS Shortfall for Examples of Algebraic, LDPC and Turbo Codes

The first example is the extended BCH code (128, 99, 10) whose coefficients up to weight 30 of the weight enumerator polynomial [5] are tabulated in Table 14.2.

Table 14.2 Low-weight spectral terms for the extended BCH (128, 99) code

Weight	A_d
0	1
10	796544
12	90180160
14	6463889536
16	347764539928
18	14127559573120
20	445754705469248
22	11149685265467776
24	224811690627712384
26	3704895377802191104
28	50486556173121673600
30	574502176730571255552

The PDF of the number of erased bits that are correctable up to the maximum of 29 erasures, derived from Eq. (14.1), is shown plotted in Fig. 14.4. Also shown plotted in Fig. 14.4 is the performance obtained numerically. It is straightforward, by computer simulation, to evaluate the erasure correcting performance of the code by generating a pattern of erasures randomly and solving these in turn using the parity-check equations. This procedure corresponds to maximum likelihood (ML) decoding [6, 17]. Moreover, the codeword responsible for any instances of non-MDS performance, (due to this erasure pattern) can be determined by back substitution into the solved parity-check equations. Except for short codes or very high rate codes, it is not possible to complete this procedure exhaustively, because there are too many combinations of erasure patterns. For example, there are 4.67×10^{28} combinations of 29 erasures in this code of length 128 bits. In contrast, there are relatively few low-weight codewords responsible for the non-MDS performance of the code. For example, each codeword of weight 10 is responsible for $\binom{118}{19} = 4.13 \times 10^{21}$ erasures patterns not being solvable.

As the d_{min} of this code is 10, the code is guaranteed to correct any erasure pattern containing up to 9 erasures. It can be seen from Fig. 14.4 that the probability of not being able to correct any pattern of 10 erasures is less than 10^{-8}. The probability of correcting 29 erasures, the maximum number, is 0.29. The average number of erasures corrected is 27.44, almost three times the d_{min}, and the average shortfall from MDS performance is 1.56 erased bits. The prediction of performance by the lower bound is pessimistic due to double codeword counting in erasure patterns featuring more than 25 bits or so. The effect of this is evident in Fig. 14.4. The lower bound average number of erasures corrected is 27.07, and the shortfall from MDS performance is 1.93 erasures, an error of 0.37 erasures. The erasure performance evaluation by simulation is complementary to the analysis using the weight distribution of the code, in that the simulation, being a sampling procedure, is inaccurate for short, uncorrectable erasure patterns, because few codewords are responsible for the performance in this region. For short, uncorrectable erasure patterns, the lower bound analysis is tight in this region because it not possible for these erasure patterns to contain more than one codeword due to codewords differing by at least d_{min}.

The distribution of the codeword weights responsible for non-MDS performance of this code is shown in Fig. 14.5.

This is in contrast to the distribution of low-weight codewords shown in Fig. 14.6. Although there are a larger number of higher weight codewords, there is less chance of an erasure pattern containing a higher weight codeword. The maximum occurrence is for weight 14 codewords as shown in Fig. 14.5.

The FER performance of the BCH (128, 107, 10) code is shown plotted in Fig. 14.7 as a function of relative capacity defined by $\frac{(1-p)n}{k}$. Also, plotted in Fig. 14.7 is the FER performance of a hypothetical (128, 99, 30) MDS code. Equations (14.15) and (14.14), respectively, were used to derive Fig. 14.7. As may be seen from Fig. 14.7, there is a significant shortfall in capacity even for the optimum MDS code. This shortfall is attributable to the relatively short length of the code. At 10^{-9} FER, the BCH (128, 99, 10) code achieves approximately 80% of the erasure channel capacity.

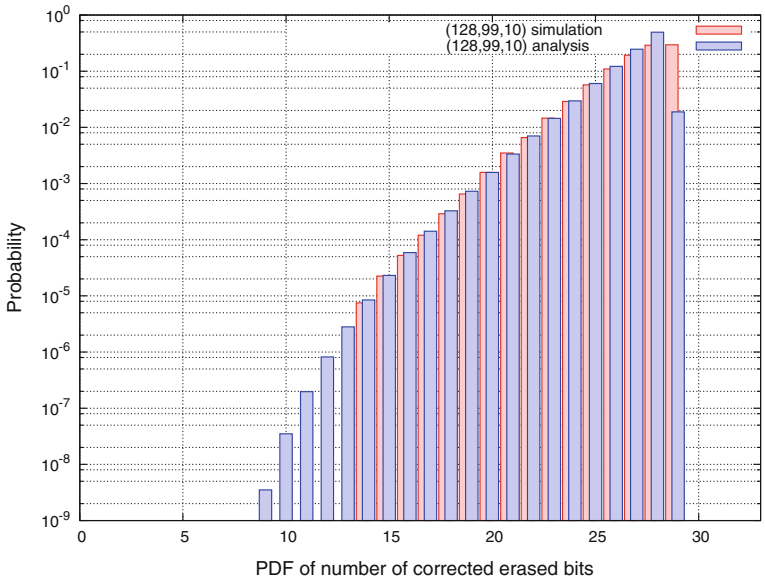

Fig. 14.4 Erasure performance for the (128, 99, 10) Extended BCH Code

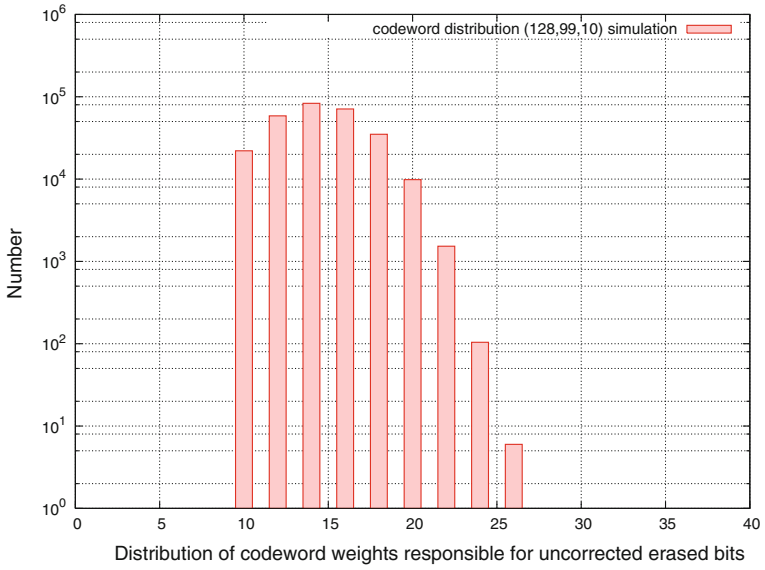

Fig. 14.5 Distribution of codeword weights responsible for non-MDS performance, of the (128, 99, 10) BCH Code

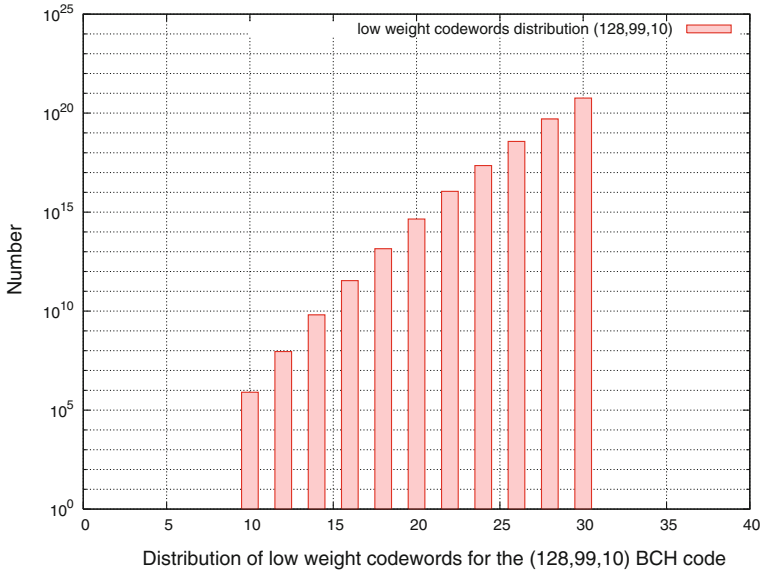

Fig. 14.6 Distribution of low-weight codewords for the (128, 99, 10) BCH code

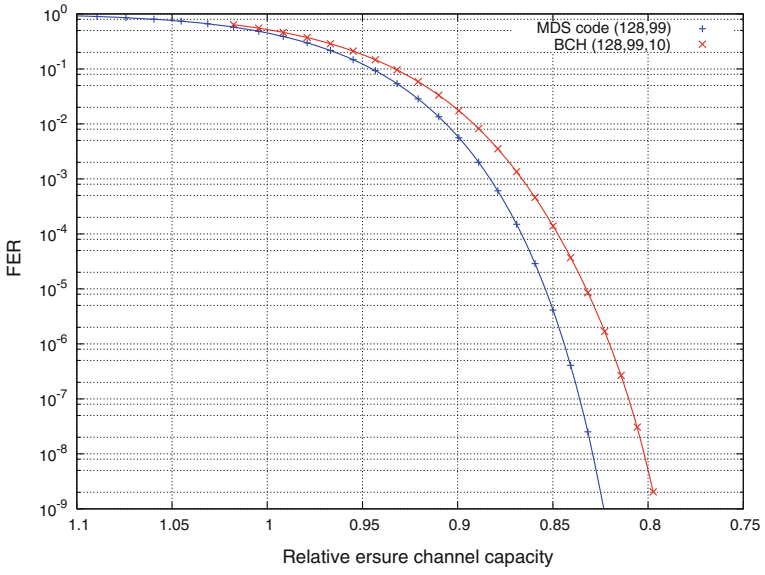

Fig. 14.7 FER performance for the (128, 99, 10) BCH code for the erasure channel

Table 14.3 Spectral terms up to weight 50 for the extended BCH (256, 207) code

Weight	A_d
0	1
14	159479040
16	36023345712
18	6713050656000
20	996444422768640
22	119599526889384960
24	11813208348266177280
26	973987499253055749120
28	67857073021007558686720
30	4036793565003066065373696
32	206926366333597318696425720
34	9212465086525810564304939520
36	358715843060045310259622139904
38	12292268362368552720093779880960
40	372755158433879986474102933212928
42	10052700091541303286178365979008000
44	242189310556445744774611488568535040
46	5233629101357641331155176578460897024
48	101819140628807204943892435954902207120
50	1789357109760781792970450788764603959040

The maximum capacity achievable by any (128, 99) binary code as represented by a (128, 99, 30) MDS code is approximately 82.5%.

An example of a longer code is the (256, 207, 14) extended BCH code. The coefficients up to weight 50 of the weight enumerator polynomial [10] are tabulated in Table 14.3. The evaluated erasure correcting performance of this code is shown in Fig. 14.8, and the code is able to correct up to 49 erasures. It can be seen from Fig. 14.8 that there is a close match between the lower bound analysis and the simulation results for the number of erasures between 34 and 46. Beyond 46 erasures, the lower bound becomes increasingly pessimistic due to double counting of codewords. Below 34 erasures the simulation results are erratic due to insufficient samples. It can be seen from Fig. 14.8 that the probability of correcting only 14 erasures is less than 10^{-13} (actually 5.4×10^{-14}) even though the d_{min} of the code is 14. If a significant level of erasure correcting failures is defined as 10^{-6}, then from Fig. 14.8, this code is capable of correcting up to 30 erasures even though the guaranteed number of correctable erasures is only 13. The average number of erasures correctable by the code is 47.4, an average shortfall of 1.6 erased bits. The distribution of codeword weights responsible for the non-MDS performance of this code is shown in Fig. 14.9.

The FER performance of the BCH (256, 207, 14) code is shown plotted in Fig. 14.10 as a function of relative capacity defined by $\frac{(1-p)n}{k}$. Also plotted in

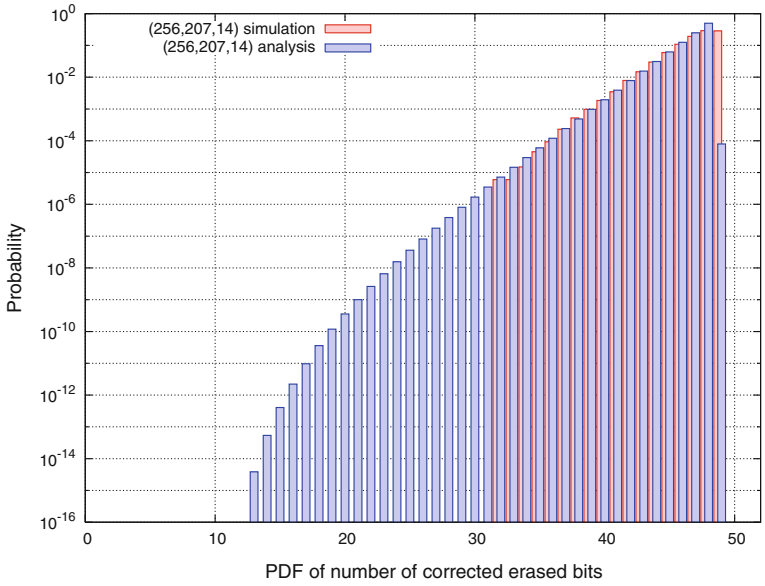

Fig. 14.8 PDF of erasure corrections for the (256, 207, 14) Extended BCH Code

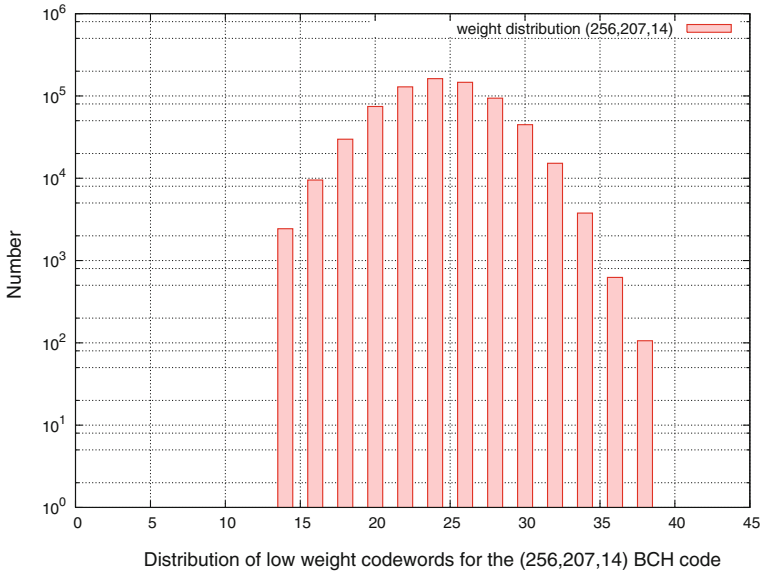

Fig. 14.9 Distribution of codeword weights responsible for non-MDS performance, for the extended (256, 207, 14) BCH Code

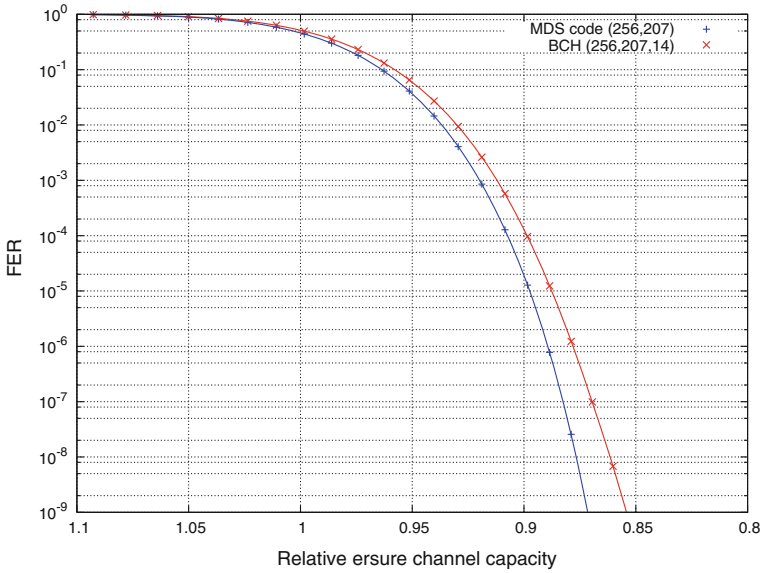

Fig. 14.10 FER performance for the (256, 207, 14) BCH Code for the erasure channel

Fig. 14.10 is the FER performance of a hypothetical (256, 207, 50) MDS code. Equations (14.15) and (14.14), respectively, were used to derive Fig. 14.10. As may be seen from Fig. 14.10, there is less of a shortfall in capacity compared to the BCH (128, 107, 10) code. At 10^{-9} FER, the BCH (256, 207, 14) code achieves approximately 85.5% of the erasure channel capacity. The maximum capacity achievable by any (256, 207) binary code as represented by the (256, 207, 50) hypothetical MDS code is approximately 87%.

The next code to be investigated is the (512, 457, 14) extended BCH code which was chosen because it is comparable to the (256, 207, 14) code in being able to correct a similar maximum number of erasures (55 *cf.* 49) and has the same d_{min} of 14. Unfortunately, the weight enumerator polynomial has yet to be determined, and only erasure simulation results may be obtained. Figure 14.11 shows the performance of this code. The average number of erasures corrected is 53.4, an average shortfall of 1.6 erased bits. The average shortfall is identical to the (256, 207, 14) extended BCH code. Also, the probability of achieving MDS code performance, i.e. being able to correct all $n - k$ erasures is also the same and equal to 0.29. The distribution of codeword weights responsible for non-MDS performance of the (512, 457, 14) code is very similar to that for the (256, 207, 14) code, as shown in Fig. 14.12.

An example of an extended cyclic quadratic residue code is the (168, 84, 24) code whose coefficients of the weight enumerator polynomial have been recently determined [20] and are tabulated up to weight 72 in Table 14.4. This code is a self-dual, doubly even code, but not extremal because its d_{min} is not 32 but 24 [3]. The FER performance of the (168, 84, 24) code is shown plotted in Fig. 14.13 as

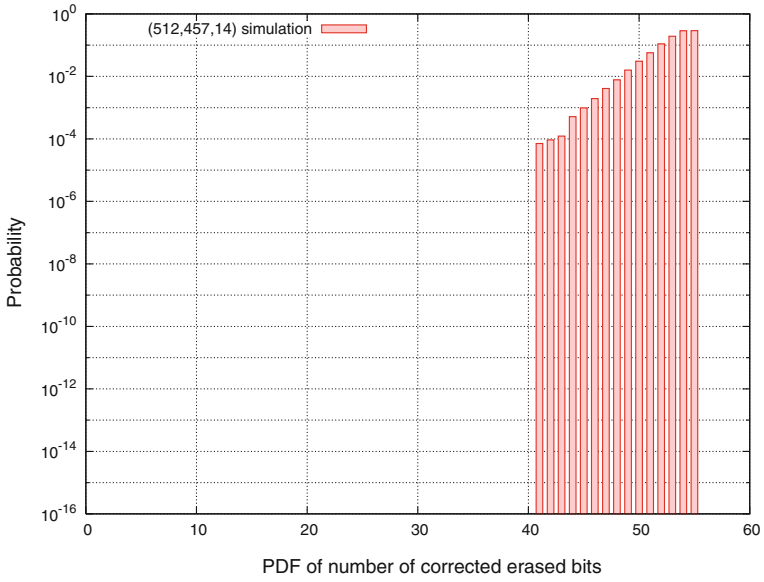

Fig. 14.11 PDF of erasure corrections for the (512, 457, 14) Extended BCH Code

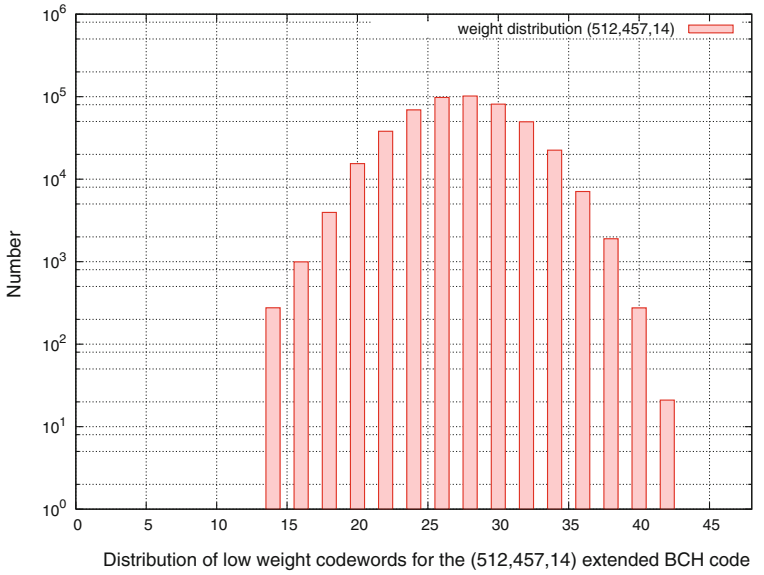

Fig. 14.12 Distribution of codeword weights responsible for non-MDS performance, for the extended (512, 457, 14) BCH Code

Table 14.4 Spectral terms up to weight 72 for the extended Quadratic Residue (168, 84) code

Weight	A_d
0	1
24	776216
28	18130188
32	5550332508
36	1251282702264
40	166071600559137
44	13047136918828740
48	629048543890724216
52	19087130695796615088
56	372099690249351071112
60	4739291519495550245228
64	39973673337590380474086
68	2256966777271886905701 84
72	860241108921860741947676

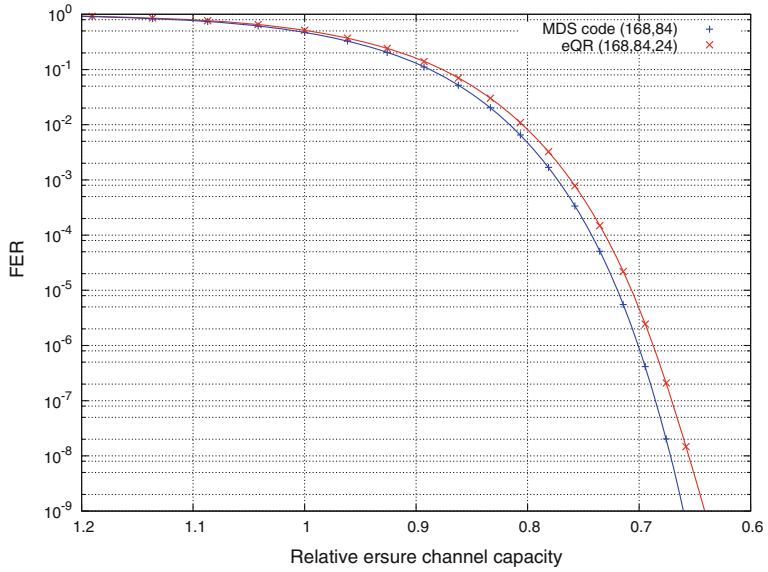

Fig. 14.13 FER performance for the (168, 84, 24) eQR Code for the erasure channel

a function of relative capacity defined by $\frac{(1-p)n}{k}$. Also plotted in Fig. 14.13 is the FER performance of a hypothetical (168, 84, 85) MDS code. Equations (14.15) and (14.14), respectively, were used to derive Fig. 14.13. The performance of the (168, 84, 24) code is close to that of the hypothetical MDS code but both codes are around 30% from capacity at 10^{-6} FER.

The erasure correcting performance of non-algebraic designed codes is quite different from algebraic designed codes as may be seen from the performance results for a (240, 120, 16) turbo code shown in Fig. 14.14. The turbo code features memory 4 constituent recursive encoders and a code matched, modified S interleaver, in order to maximise the d_{min} of the code. The average number of erasures correctable by the code is 116.5 and the average shortfall is 3.5 erased bits. The distribution of codeword weights responsible for non-MDS performance of the (240, 120, 16) code is very different from the algebraic codes and features a flat distribution as shown in Fig. 14.15.

Similarly, the erasure correcting performance of a (200, 100, 11) LDPC code designed using the Progressive Edge Growth (PEG) algorithm [12] is again quite different from the algebraic codes as shown in Fig. 14.16. As is typical of randomly generated LDPC codes, the d_{min} of the code is quite small at 11, even though the code has been optimised. For this code, the average number of correctable erasures is 93.19 and the average shortfall is 6.81 erased bits. This is markedly worse than the turbo code performance. It is the preponderance of low-weight codewords that is responsible for the inferior performance of this code compared to the other codes as shown by the codeword weight distribution in Fig. 14.17.

The relative weakness of the LDPC code and turbo code becomes clear when compared to a good algebraic code with similar parameters. There is a (200, 100, 32) extended quadratic residue code. The p.d.f. of the number of erasures corrected by this code is shown in Fig. 14.18. The difference between having a d_{min} of 32 compared to 16 for the turbo code and 10 for the LDPC code is dramatic. The average number of correctable erasures is 98.4 and the average shortfall is 1.6 erased bits. The weight enumerator polynomial of this self-dual code, is currently unknown as evaluation of the 2^{100} codewords is currently beyond the reach of today's computers. However, the distribution of codeword weights responsible for non-MDS performance of the (200, 100, 32) code which is shown in Fig. 14.19 indicates the doubly even codewords of this code and the d_{min} of 32.

14.5.1 Turbo Codes with Dithered Relative Prime (DRP) Interleavers

DRP interleavers were introduced in [4]. They have been shown to produce some of the largest minimum distances for turbo codes. However, the iterative decoding algorithm does not exploit this performance to its full on AWGN channels where the performance of these interleavers is similar to that of randomly designed interleavers having lower minimum distance. This is due to convergence problems in the error floor region. A DRP interleaver is a concatenation of 3 interleavers, the two dithers A, B and a relative prime interleaver π:

$$I(i) = B(\pi(A(i))) \tag{14.26}$$

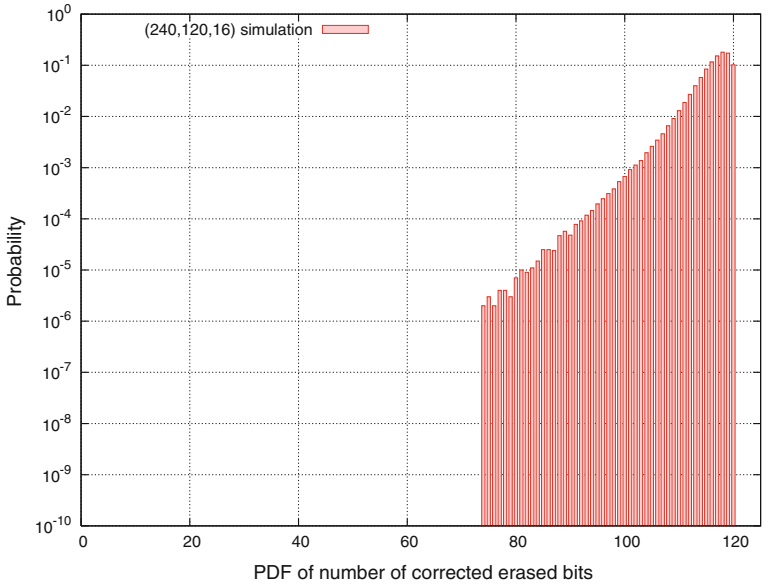

Fig. 14.14 PDF of erasure corrections for the (240, 120, 16) turbo code

Fig. 14.15 Distribution of codeword weights responsible for non-MDS performance, for the (240, 120, 16) turbo code

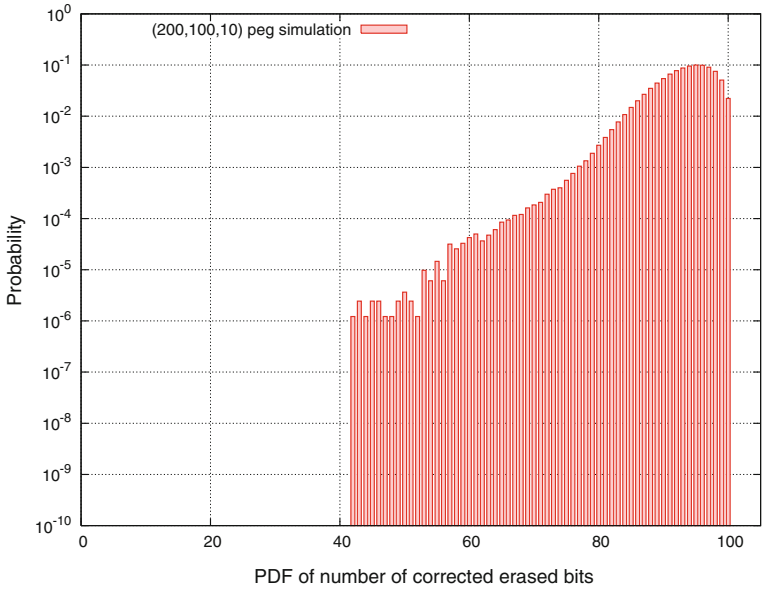

Fig. 14.16 PDF of erasure corrections for the (200, 100, 10) PEG LDPC code

Fig. 14.17 Distribution of codeword weights responsible for non-MDS performance, for the (200, 100, 10) PEG LDPC code

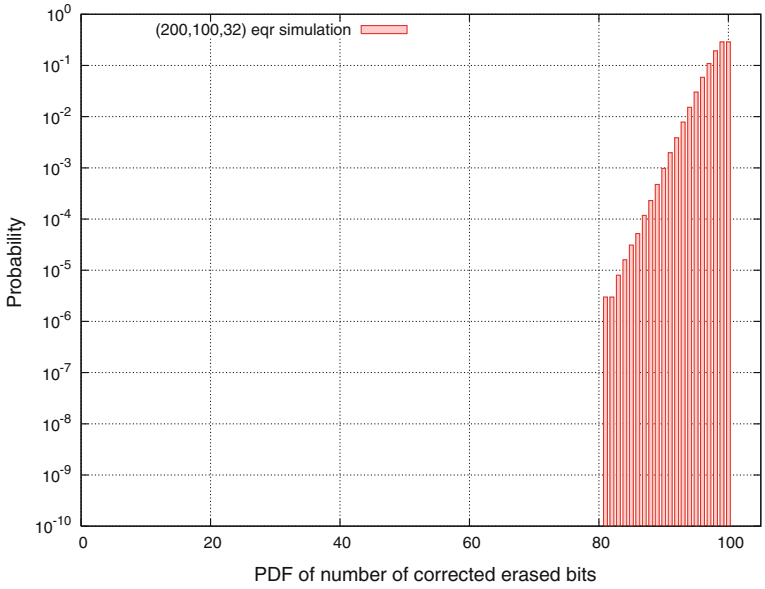

Fig. 14.18 PDF of erasure corrections for the (200, 100, 32) Extended QR Code

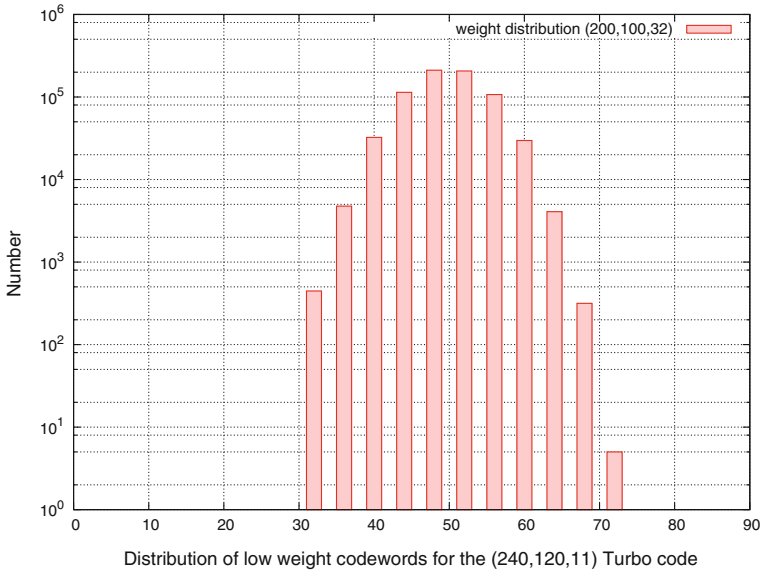

Fig. 14.19 Distribution of codeword weights responsible for non-MDS performance, for the (200, 100, 32) Extended QR Code

Table 14.5 Minimum distance of turbo codes using DRP interleavers as compared to S-random interleavers

k	40	200	400	1000
DRP	19	33	38	45
S-RAN	13	17	22	26

The dithers are short permutations, generally of length $m = 4, 8, 16$ depending on the length of the overall interleaver. We have

$$A(i) = m\lfloor i/m \rfloor + a_{i\%m} \tag{14.27}$$

$$B(i) = m\lfloor i/m \rfloor + b_{i\%m} \tag{14.28}$$

$$\pi(i) = (pi + q)\%m, \tag{14.29}$$

where a, b, are permutations of length m and p must be relatively prime to k. If a, b and p are properly chosen, the minimum distance of turbo codes can be drastically improved as compared to that of a turbo code using a typical $\mathscr{S}$-random interleaver. A comparison is shown in Table 14.5 for memory 3 component codes.

As an example two turbo codes are considered, one employing a DRP interleavers, having parameters (120, 40, 19) and another employing a typical $\mathscr{S}$-random interleaver and having parameters (120, 40, 13).

14.5.2 Effects of Weight Spectral Components

The weight spectrum of each of the two turbo codes has been determined exhaustively from the G matrix of each code by codeword enumeration using the revolving door algorithm. The weight spectrum of both of the two turbo codes is shown in Table 14.6. It should be noted that as the codes include the all ones codeword, $A_{n-j} = A_j$, only weights up to A_{60} are shown in Table 14.6.

Using the weight spectrum of each code the upper and lower bound cumulative distributions and corresponding density functions have been derived using Eqs. (14.2) and (14.3), respectively, and are compared in Fig. 14.20. It can be observed that the DRP interleaver produces a code with a significantly smaller probability of failing to correct a given number of erasures.

The MDS shortfall for the two codes is:

$$\text{MDS}_{\text{shortfall}}(120, 40, 19) = 2.95 \text{ bits} \tag{14.30}$$

$$\text{MDS}_{\text{shortfall}}(120, 40, 13) = 3.29 \text{ bits} \tag{14.31}$$

The distribution of the codeword weights responsible for the MDS shortfalls is shown in Fig. 14.21. For interest, also shown in Fig. 14.21 is the distribution for

Table 14.6 Weight spectrum of the (120, 40, 19) and (120, 40, 13) turbo codes. Multiplicity for weights larger than 60 satisfy $A_{60-i} = A_{60+i}$

Weight	Multiplicity	
	(120, 40, 19)	(120, 40, 13)
0	1	1
13	0	3
14	0	6
15	0	3
16	0	15
17	0	21
18	0	17
19	10	52
20	100	82
21	130	136
22	300	270
23	450	462
24	880	875
25	1860	2100
26	3200	3684
27	7510	7204
28	14715	15739
29	29080	30930
30	63469	64602
31	137130	137976
32	279815	279700
33	611030	608029
34	1313930	1309472
35	2672760	2671331
36	5747915	5745253
37	12058930	12045467
38	24137345	24112022
39	49505760	49486066
40	97403290	97408987
41	183989250	184005387
42	347799180	347810249
43	626446060	626489895
44	1086030660	1086006724
45	1855409520	1855608450
46	3021193870	3021448047
47	4744599030	4744412946
48	7286393500	7286669468
49	10691309800	10690683197
50	15157473609	15156479947

(continued)

Table 14.6 (continued)

Weight	Multiplicity	
	(120, 40, 19)	(120, 40, 13)
51	20938289040	20939153481
52	27702927865	27702635729
53	35480878330	35481273341
54	44209386960	44210370096
55	52854740864	52853468145
56	61256875090	61257409658
57	69008678970	69008947092
58	74677319465	74677092916
59	78428541430	78428875230
60	80007083570	80006086770

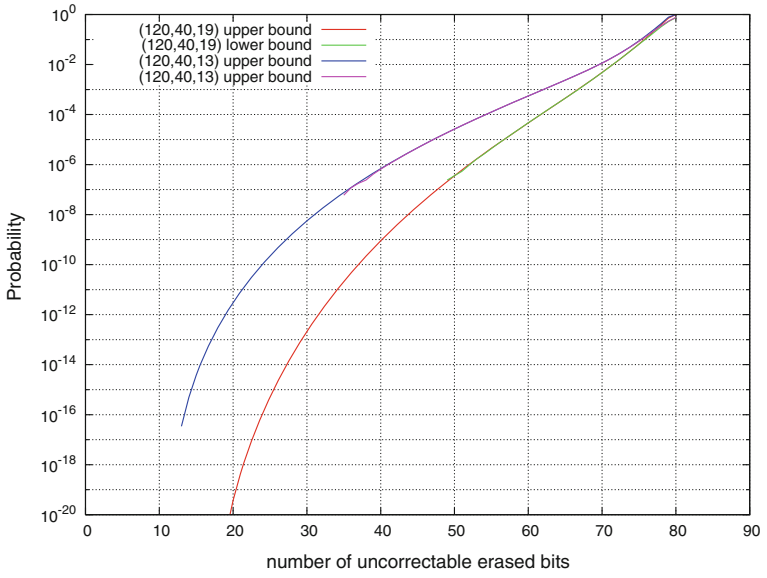

Fig. 14.20 Probability of Maximum Likelihood decoder failure

(120, 40, 28) best known linear code. This code, which is chosen to have the same block length and code rate as the turbo code, is derived by shortening a (130, 50, 28) code obtained by adding two parity checks to the (128, 50, 28) extended BCH. This linear code has an MDS shortfall of 1.62 bits and its weight spectrum consists of doubly even codewords as shown in Table 14.7. For the turbo codes the contribution made by the lower weight codewords is apparent in Fig. 14.21, and this is confirmed by the plot of the cumulative contribution made by the lower weight codewords shown in Fig. 14.22.

Table 14.7 Weight spectrum of the linear (120, 40, 28) code derived from the extended BCH (128, 50, 28) code

Weight j	Multiplicity A_j
0	1
28	5936
32	448563
36	17974376
40	379035818
44	4415788318
48	29117944212
52	110647710572
56	245341756158
60	319670621834
64	245340760447
68	110648904336
72	29117236550
76	4415980114
80	379051988
84	17949020
88	453586
92	5910
96	37

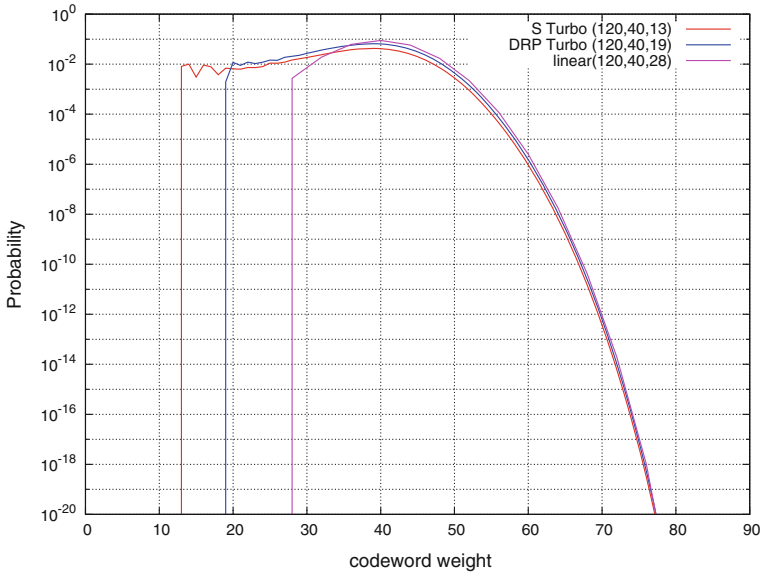

Fig. 14.21 Distribution of codeword weights responsible for non-MDS performance

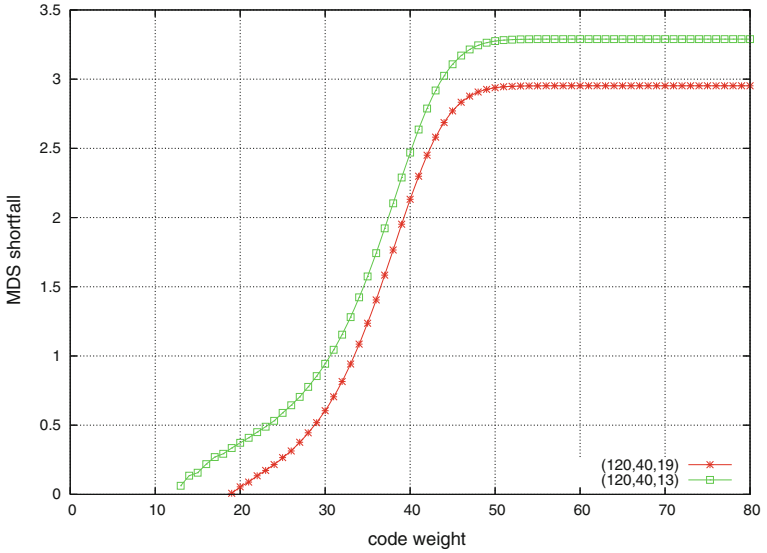

Fig. 14.22 Cumulative code weight contribution to MDS shortfall

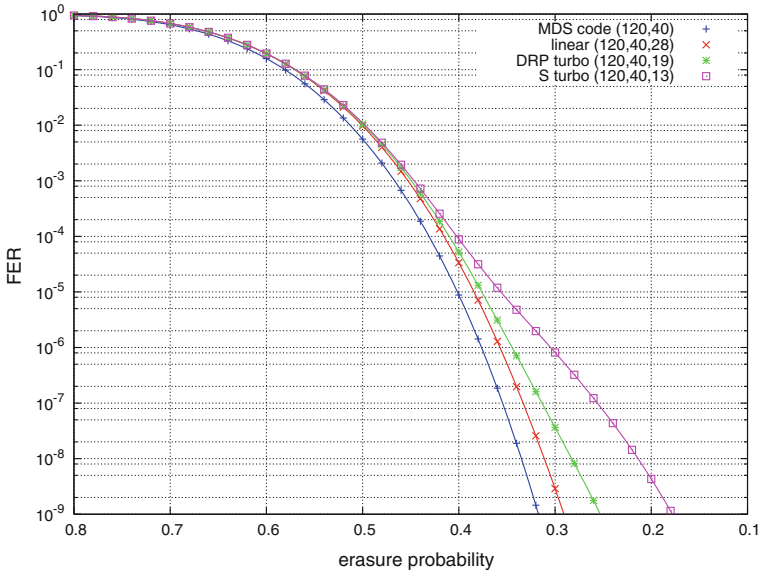

Fig. 14.23 Probability of ML decoder error for the erasure channel

For the erasure channel, the performance of the two turbo codes and the (120, 40, 28) code is given by (14.15) and is shown in Fig. 14.23 assuming ML decoding. Also shown in Fig. 14.23 is the performance of a (hypothetical) binary (120, 40, 81) MDS which is given by the second term of (14.15). The code derived from the shortened, extended BCH code, (120, 40, 28), has the best performance and compares well to the lower bound provided by the MDS hypothetical code. The DRP interleaver turbo code also has good performance, but the $\mathscr{S}$-random interleaver turbo code shows an error floor due to the d_{min} of 13.

14.6 Determination of the d_{min} of Any Linear Code

It is well known that the determination of weights of any linear code is a Nondeterministic Polynomial time (NP) hard problem [8] and except for short codes, the best methods for determining the minimum Hamming distance, d_{min} codeword of a linear code, to date, are probabilistically based [2]. Most methods are based on the generator matrix, the G matrix of the code and tend to be biased towards searching using constrained information weight codewords. Such methods become less effective for long codes or codes with code rates around $\frac{1}{2}$ because the weights of the evaluated codewords tend to be binomially distributed with average weight $\frac{n}{2}$ [15].

Corollary 2 from Sect. 14.2 above, provides the basis of a probabilistic method to find low-weight codewords in a significantly smaller search space than the G matrix methods. Given an uncorrectable erasure pattern of $n-k$ erasures, from Corollary 2, the codeword weight is less than or equal to $n-k$. The search method suggested by this, becomes one of randomly generating erasure patterns of $n-k+1$ erasures, which of course are uncorrectable by any (n,k) code, and determining the codeword and its weight from (14.2). This time, the weights of the evaluated codewords will tend to be binomially distributed with average weight $\frac{n-k+1}{2}$. With this trend, for N_{trials} the number of codewords determined with weight d, M_d is given by

$$M_d = N_{trials} \frac{(n-k+1)!}{d!(n-k-d+1)!2^{n-k+1}} \tag{14.32}$$

As an example of this approach, the self-dual, bordered, double-circulant code (168, 84) based on the prime number 83, is described in [11] as having an unconfirmed d_{min} of 28. From (14.32) when using 18,000 trials, 10 codewords of weight 28 will be found on average. However, as the code is doubly even and only has codewords weights which are a multiple of 4, using 18,000 trials, 40 codewords are expected. In a set of trials using this method for the (168, 84) code, 61 codewords of weight 28 were found with 18,000 trials. Furthermore, 87 codewords of weight 24 were also found indicating that the d_{min} of this code is 24 and not 28 as was originally expected in [11].

The search method can be improved by biasing towards the evaluation of erasure patterns that have small numbers of erasures that cannot be solved. Recalling the

analysis in Sect. 14.2, as the parity-check equations are Gaussian reduced, no erased bit is a function of any other erased bits. There will be $n - k - s$ remaining parity-check equations, which do not contain the erased bit coordinates $\mathbf{x_f}$. These remaining equations may be searched to see if there is an unerased bit coordinate, that is not present in any of the equations. If there is one such coordinate, then this coordinate in conjunction with the erased coordinates solved so far forms an uncorrectable erasure pattern involving only s erasures instead of $n - k + 1$ erasures. With this procedure, biased towards small numbers of unsolvable erasures, it was found that, for the above code, 21 distinct codewords of weight 24 and 17 distinct codewords of weight 28 were determined in 1000 trials and the search took approximately 2 s on a typical 2.8GHz, Personal Computer (PC).

In another example, the (216, 108) self dual, bordered double-circulant code is given in [11] with an unconfirmed d_{min} of 36. With 1000 trials which took 7 s on the PC, 11 distinct codewords were found with weight 24 and a longer evaluation confirmed that the d_{min} of this code is indeed 24.

14.7 Summary

Analysis of the erasure correcting performance of linear, binary codes has provided the surprising result that many codes can correct, on average, almost $n-k$ erasures and have a performance close to the optimum performance as represented by (hypothetical), binary MDS codes. It was shown that for codes having a weight distribution approximating to a binomial distribution, and this includes many common codes, such as BCH codes, Goppa codes and self-dual codes, that these codes can correct at least $n - k - 2$ erasures on average, and closely match the FER performance of MDS codes as code lengths increase. The asymptotic performance achieves capacity for the erasure channel. It was also shown that codes designed for iterative decoders, the turbo and LDPC codes, are relatively weak codes for the erasure channel and compare poorly with algebraically designed codes. Turbo codes, designed for optimised d_{min}, were found to outperform LDPC codes.

For turbo codes using DRP interleavers for the erasure channel using ML decoding, the result is that these relatively short turbo codes are (on average), only about 3 erasures away from optimal MDS performance. The decoder error rate performance of the two turbo codes when using ML decoding on the erasure channel was compared to (120, 40, 28) best known linear code and a hypothetical binary MDS code. The DRP interleaver demonstrated a clear advantage over the $\mathscr{S}$-random interleaver and was not too far way from MDS performance. Analysis of the performance of longer turbo codes is rather problematic.

Determination of the erasure correcting performance of a code provides a means of determining the d_{min} of a code and an efficient search method was described. Using the method, the d_{min} results for two self-dual codes, whose d_{min} values were previously unknown were determined, and these codes were found to be (168, 84, 24) and (216, 108, 24) codes.

References

1. Berrou, C., Glavieux, A., Thitimajshima, P.: Near Shannon limit error-correcting coding: Turbo codes. In: Proceedings of the IEEE International Conference on Communications, Geneva, Switzerland, pp. 1064–1070 (1993)
2. Canteaut, A., Chabaud, F.: A new algorithm for finding minimum weight words in a linear code: application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. IEEE Trans. Inf. Theory **44**(1), 367–378 (1998)
3. Conway, J.H., Sloane, N.J.A.: A new upper bound on the minimum distance of self-dual codes. IEEE Trans. Inf. Theory **36**(6), 1319–1333 (1990)
4. Crozier, S., Guinard, P.: Distance upper bounds and true minimum distance results for Turbo codes designed with DRP interleavers. In: Proceedings of the 3rd International Symposium on Turbo Codes, pp. 169–172 (2003)
5. Desaki, Y., Fujiwara, T., Kasami, T.: The weight distribution of extended binary primitive BCH code of length 128. IEEE Trans. Inf. Theory **43**(4), 1364–1371 (1997)
6. Di, C., Proietti, D., Telatar, I.E., Richardson, T.J., Urbanke, R.L.: Finite-length analysis of low-density parity-check codes on the binary erasure channel. IEEE Trans. Inf. Theory **48**(6), 1570–1579 (2002)
7. Dumer, I., Farrell, P.: Erasure correction performance of linear block codes. In: Cohen, G., Litsyn, S., Lobstein, A., Zemor, G. (eds.) Lecture Notes in Computer Science, vol. 781, pp. 316–326. Springer, Berlin (1993)
8. Dumer, I., Micciancio, D., Sudan, M.: Hardness of approximating the minimum distance of a linear code. IEEE Trans. Inf. Theory **49**(1), 22–37 (2003)
9. Elias, P.: Coding for two noisy channels. Third London Symposium on Information Theory. Academic Press, New York (1956)
10. Fujiwara, T., Kasami, T.: The weight distribution of (256, k) extended binary primitive BCH code with $k \leq 63$, $k \geq 207$. Technical Report of IEICE IT97-46:29–33 (1997)
11. Gulliver, T.A., Senkevitch, N.: On a class of self-dual codes derived from quadratic residues. IEEE Trans. Inf. Theory **45**(2), 701–702 (1999)
12. Hu, X.Y., Eleftheriou, E., Arnold, D.M.: Irregular progressive edge-growth tanner graphs. In: Proceedings of IEEE International Symposium on Information Theory (ISIT), Lausanne, Switzerland (2002)
13. Krasikov, I., Litsyn, S.: On spectra of BCH codes. IEEE Trans. Inf. Theory **41**(3), 786–788 (1995)
14. Krasikov, I., Litsyn, S.: On the accuracy of the binomial approximation to the distance distribution of codes. IEEE Trans. Inf. Theory **41**(5), 1472–1474 (1995)
15. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland, Amsterdam (1977)
16. Peterson, W.: Error-Correcting Codes. MIT Press, Cambridge (1961)
17. Pishro-Nik, H., Fekri, F.: On decoding of low-density parity-check codes over the binary erasure channel. IEEE Trans. Inf. Theory **50**(3), 439–454 (2004)
18. Rizzo, L.: Effective erasure codes for reliable computer communication protocols. ACM SIGCOMM Comput. Commun. Rev. **27**(2), 24–36 (1997)
19. Roychowdhury, V.P., Vatan, F.: Bounds for the weight distribution of weakly self-dual codes. IEEE Trans. Inf. Theory **47**(1), 393–396 (2001)
20. Tjhai C., Tomlinson M., Horan R., Ahmed M., Ambroze M.: On the efficient codewords counting algorithm and the weight distribution of the binary quadratic double-circulant codes. In: Proceedings of IEEE Information Theory Workshop, Chengdu, China, 22–26 Oct. 2006, pp. 42–46 (2006)